Docket No. 042390.P10447
Express Mail No. EM051375395US

UNITED STATES PATENT APPLICATION

FOR

A METHOD FOR SECURELY USING A SINGLE PASSWORD FOR MULTIPLE PURPOSES

INVENTORS:

Keen W. Chan
Ernest F. Brickell

PREPARED BY:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 Wilshire Blvd., 7th Floor
Los Angeles, CA 90025-1026
(310) 207-3800

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

This invention relates to application password protection, and more particularly to a method of generating a plurality of passwords from a single strong password or passphrase that is not stored.

BACKGROUND INFORMATION

Many of todays software applications require users to enter a password. These applications may reside on a personal computer and/or a server that may be connected to a personal computer via a network. If a user uses a plurality of software applications that require the use of a password to gain entry, the user must either memorize many passwords, have these many passwords stored, or write them down. The passwords may be stored on the personal computer and/or the server where the software application resides. Some of these stored passwords may not be encrypted. Many users use the same or similar passwords, or a set of passwords to ease the memory requirement. A problem with this scheme is that not all software applications use cryptographic means of protecting the passphrase, or secret that is entered by the user. Thus, the users passwords may be compromised.

Also, a compromise of a password (also known as a passphrase) to one application may allow all other applications that use the same password to be compromised. Further, the password is sometimes used in software applications, such as wrapping a cryptographic key, in which the user can be attacked with a brute force password search. For example, an adversary that obtains a wrapped cryptographic key can test if the correct password is

found. Thus, if the password size is not too large, the adversary can search over all of the password space until the correct password is found. To protect against this theft, a large or complex password is necessary. The problem with having large complex passwords is that a user must now remember a plurality of long or complex passwords.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to "an" or "one" embodiment in this disclosure are not necessarily to the same embodiment, and such references mean at least one.

Figure 1 illustrates an embodiment of the invention having a client password generator.

Figure 2 illustrates an embodiment of the invention having a table entry.

Figure 3 illustrates an embodiment of the invention having a graphical user interface (GUI) for entering user information.

Figure 4 illustrates an embodiment of the invention having a GUI for entering user information and selecting an application.

Figure 5 illustrates an embodiment of the invention having a GUI for entering user information, selecting an application and dissplaying a generated password.

Figure 6 illustrates a flow diagram of a process of an embodiment of the invention.

Figure 7 illustrates a flow diagram of a process of an embodiment of the invention which also determines if a new strong password or passphrase is needed.

DETAILED DESCRIPTION

The invention generally relates to a method to allow application software users to use a single strong password or passphrase that is used to generate a plurality of individual passwords or passphrases for a plurality of software applications. Referring to the figures, exemplary embodiments of the invention will now be described. The exemplary embodiments are provided to illustrate the invention and should not be construed as limiting the scope of the invention.

Figure 1 illustrates an embodiment of the invention comprising personal computer 100, client password generator 110, server 120, server cryptographic support services 150, client software manufacturer/developer 130, and software application(s) 115 and 140. Software application 115 and 140 can be any software application requiring a password to enable entry. Software application 115 can run on server 120, where software application 140 can run on personal computer 100. It should be noted that software application 140 can also be run on other devices, such as personal digital assistants (PDAs), cellular telephones, and other similar portable devices. There may be many software applications 140 to be run on personal computer 100, and likewise for software application 115 on server 120. Client software application manufacturer 130 develops versions of client password generator 110.

In one embodiment, client password generator 110 contains random number generator (RNG) 205 to generate a salt value. RNG 205 may be any conventional RNG. A salt value may be generated for each software application. **Figure 2** illustrates table entry 200 having salt value entries 210

and application entries 220. Each application that uses a generated password (generated by client password generator 110) has an associated salt value entry 210. In another embodiment, salt value entries 210 are random values selected by each client software application developer 130.

In one embodiment, client password generator prompts a user for a user identification and a strong password or passphrase. A strong password or passphrase is a password or passphrase that has enough entropy to prevent an adversary from easily determining the strong password or passphrase through a brute force password search over possible passwords. The user identification and strong password or passphrase can be entered from devices such as a keyboard, a voice activated system, or a computer pointing device. **Figure 3** illustrates one embodiment of graphical user interface (GUI) 300 that prompts a user to enter user identification (or username) 310 and strong password or passphrase 320. **Figure 4** illustrates an embodiment that prompts a user for user identification 310, strong password or passphrase 320, and application menu 430 to select an application the user desires to generate a password for. If an application does not exist in application menu 430, a user can enter the name of the new application in an application entry GUI (not shown). **Figure 5** illustrates an embodiment that prompts a user for user identification 310, strong password or passphrase 320, application menu 430 to select an application the user desires to generate a password for, and returns a password for the selected application to generated password display 510. A user can then manually enter the generated password when prompted by an application requiring the password. In another embodiment, a plurality of

strong passwords or passphrases can be used to to generate application passwords that represent various security levels, such as confidential, secret and top secret.

In one embodiment, once client password generator 110 gets the user's strong password or passphrase entry and user identification, a salt value is retrieved for the specific application if it exists in table entry 200. If a salt value does not exist for the specific application that a user desires a password to be generated, RNG 205 generates a salt value that is entered in table entry 200 that is associated with the specific application.

Client password generator 110 uses the strong password or passphrase, user identification and salt value to generate an application specific password that is a hash of the strong password or passphrase, user identification and salt value. Standard hash programs can be used, such as Secure Hash Algorithm, SHA-1 and message digest algorithm, MD5. One should note that other hash algorithms can also be used. Once the hash is complete, an application specific password is returned. In one embodiment, the returned generated password is directly interfaced into the application requiring the password. The generated password may be temporarily stored on the platform each time the strong password or passphrase is entered and the application specific password is generated. The length of time that the generated password is temporarily stored is controlled by client password generator 110 in one embodiment. In another embodiment, a user inputs a predetermined time period for which the generated password may be stored dependent on the platform, such as personal computer 100 or server 120.

In one embodiment, a hash function is selected to be slow. This is to slow down an adversary that is trying a brute force attack to determine what password hashes to a given value. The adversary must compute the slow hash function for every password that he searches, while the real user needs to only compute the hash function for the single password that he enters. Thus, by slowing down the hash function, a user can use a password with less entropy and achieve equivalent security. One embodiment of a slow hash function is to iterate a hash function multiple times.

In one embodiment, since the amount of entropy depends on a hash function, if a hash function requires 0.1 seconds to complete on a 1 Gigahertz processor computer, at least 40 bits of entropy is sufficient to ensure protection from a brute force attack by an adversary to try to determine a password in less than 1 million days of computing on 1 Gigahertz processor computers. One should note that other sizes of entropy may be used.

In another embodiment, client password generator 110 monitors activity on the platform where client password generator 110 is running for certain activity, such as amount of application use or logon data. In one embodiment, client password generator 110 determines if a generated password is stored on a platform. If the generated password is not stored on the platform, client password generator 110 prompts a user to enter a strong password or passphrase and input information to generate the application specific password. It should be noted in one embodiment, a user can choose not to store the generated password on the platform running the client password generator. In this embodiment, client password generator 110

prompts the user for a strong password or passphrase and input information each time a password needs to be generated for an application to be entered.

Figure 6 illustrates a flow diagram of an embodiment of the invention for client password generator 110 to generate an application password. Process 600 starts at operation 605 as a user executes client password generator 110. Next, operation 610 is executed prompting a user for a strong password or passphrase and a user identification. Process 600 continues with operation 620 determining if a salt exists for the specific application. In one embodiment, operation 620 queries table entry 200 and searches for an application name and associated salt. If a salt does not exist for a selected application, operation 630 generates a salt for the selected application. After a salt is generated, in one embodiment operation 635 stores salt in table entry 200. Process 600 continues with operation 620 determining if a salt exists. At this time, process 600 continues to operation 650. Operation 650 generates an application specific password that is a hash of the user identification, strong password or passphrase and salt that is associated for the specified application. Process 600 then continues with operation 660 returning a generated password for the specified application.

Occasionally, an application requiring a password may prompt a user to change the required password, or a user may choose an option on the application to change his password. In one embodiment, the application would require the old password and a new password. To generate the old password, client password generator 110 initiates process 600 with a request

for the old password. To generate the new password, client password generator 110 initiates process 600 with a request for the new password.

Figure 7 illustrates process 700 of an embodiment of the invention that adds operations 710 and 720 to process 600 illustrated in **Figure 6**. Operation 710 determines if a new strong password or passphrase is required to be entered by a user. If a new strong password or passphrase is required, operation 720 prompts the user for a new strong password or passphrase and user identification. After the user enters this information, process 700 continues with operation 710.

In one embodiment, when the user changes his strong password, he will need to change passwords on all applications so the applications will all be using a password derived from the new strong password.

In one embodiment, an application profile is associated with each application and indicates if a new strong password or passphrase is being used for each application in table entry 200. In another embodiment, a field may be added to indicate a date a new strong password was entered. Since changing the passwords for all applications may be very time consuming, the user can choose to change some passwords at a later time. In one embodiment, client password generator 110 prompts the user for each application that has not yet used the new strong password or passphrase to generate an application specific password. The user is prompted to enter the old strong password or passphrase and then the new strong password and passphrase at this time.

In another embodiment, after the user has entered the new strong password, a computation is performed and the result stored that allows the

client password generator to compute the old strong password from the new strong password. An example of this would be to encrypt the old strong password with the new strong password. In this case, the user would not need to enter the old strong password in order to change the password for each application. When an application requires both the old password and the new password for an application, the user would enter just his new strong password, and the client password generator would compute the old strong password, and then generate both the old and new passwords for that application.

In one embodiment, the user is prompted to occasionally change the strong password or passphrase. A user can choose a predetermined period of time to be reminded to change the strong password or passphrase. The embodiments of the invention allow users to only need remember one strong password or passphrase while generating a plurality of application specific passwords. If an attacker was to compromise one password of a specific application, all other passwords for other applications are not compromised. Also, the strong password or passphrase is not stored. Also, operating systems and applications need not be modified for the above embodiments.

The above embodiments can also be stored on a device or medium and read by a machine to perform instructions. The device or medium may include a solid state memory device and/or a rotating magnetic or optical disk. The device or medium may be distributed when partitions of instructions have been separated into different machines, such as across an interconnection of computers.

While certain exemplary embodiments have been described and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not restrictive on the broad invention, and that this invention not be limited to the specific constructions and arrangements shown and described, since various other modifications may occur to those ordinarily skilled in the art.